SENG 474 - Assignment 1

Benjamin Frizzell

October 6, 2023

1 Separate Analysis

1.1 Decision Trees

We begin the analysis by studying the performance of decision trees in accurately classifying emails from the data set as spam. The features of the data set measure counts of various words expected in spam emails, as well as statistics on the number of capital characters found. In this section, the three split criterion which are implemented in sklearn are analyzed: Gini impurity, Entropy, and Log Loss. We expect the last two criterion to produce identical results. [1]

1.1.1 Pre-Pruning

It is well known that overfitting is a common issue that arises in the implementation decision trees. A simple approach to solving this problem is **pre-pruning**, in which we prescribe a lower bound to the number of samples required at a leaf node. If this limit is greater than one, the 'majority vote' classification of all data incident upon this leaf is taken as the overall classifier. Such a solution will reduce overfitting and allow for more simpler, generalizable models but will undoubtedly lead to some misclassification. Figure 1 shows the effect of prepruning:

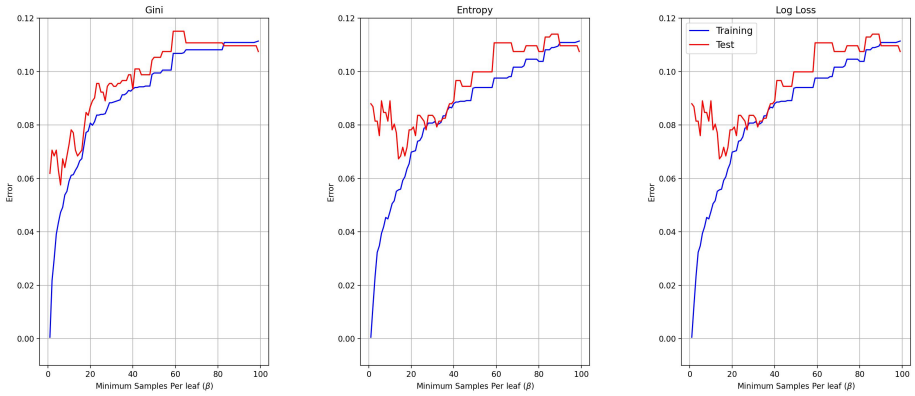

Figure 1: Training/Testing error of Decision Trees with varied pre-pruning hyperparameter

Optimal parameters were $\beta = 6$ for the Gini Impurity and $\beta = 14$ for Entropy/Log loss. We can note the effect of overfitting here; training error -but not test error- is minimized when every sample in the training set is assigned to one leaf.

1.1.2 Sample Size

Figure 2 shows the effect of reducing training sample size on the training and testing error:

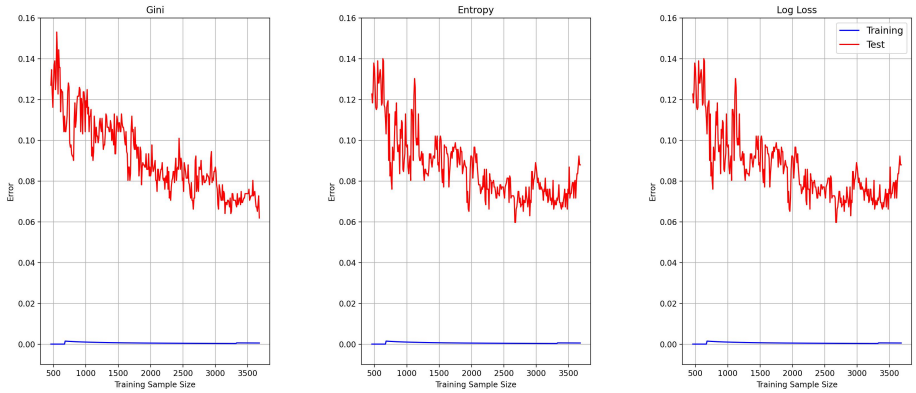

Figure 2: Training/Testing error of Decision Trees with varying training sample size

It can be noticed that while the testing error decreases with increasing training data (as expected), the training error is largely independent of the sample size, likely because pruning was not implemented. To confirm this, we repeat the above analysis and implement the optimal pre-pruning parameters obtained previously. Figure 3 shows the results.

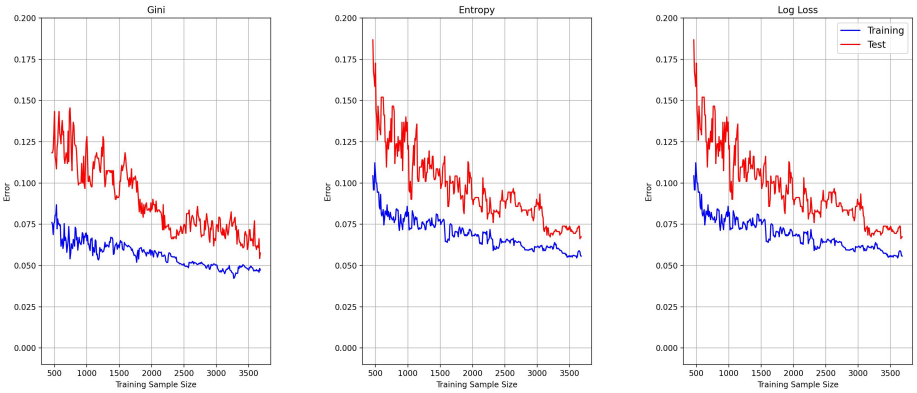

Figure 3: Training/Testing error of Decision Trees with varying training sample size,using pre-pruning.

We can see that the results are similar to Figure 2, however the overall training error has been reduced at a cost of higher testing error. This is an acceptable outcome as the generality of the model has been improved, as well as the robustness against small training sets.

1.1.3 Cost-Complexity Pruning

An alternative method of pruning is **Cost-Complexity Pruning**, in which the decision tree is grown to its full depth and pruned according to a cost complexity measure $R_\alpha(T)$ characterized by a hyperparameter α:

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}| \tag{1}$$

Where $R(T)$ is the misclassification rate of the leaves of tree T, and $|\tilde{T}|$ is the number of terminal nodes in the tree.

Cost-complexity pruning finds the root node t and it's subtree T_t with the minimal effective value of α_{eff}, where $R_{\alpha_{eff}}(t) = R_{\alpha_{eff}(T_t)}$. This can be interpreted as the subtree with the 'weakest' classification strength relative to its parent node. Subtrees are iteratively pruned until α_{eff} is greater the passed hyperparameter value. [1]

Figure 4 shows the results for varying maximum α_{eff}:

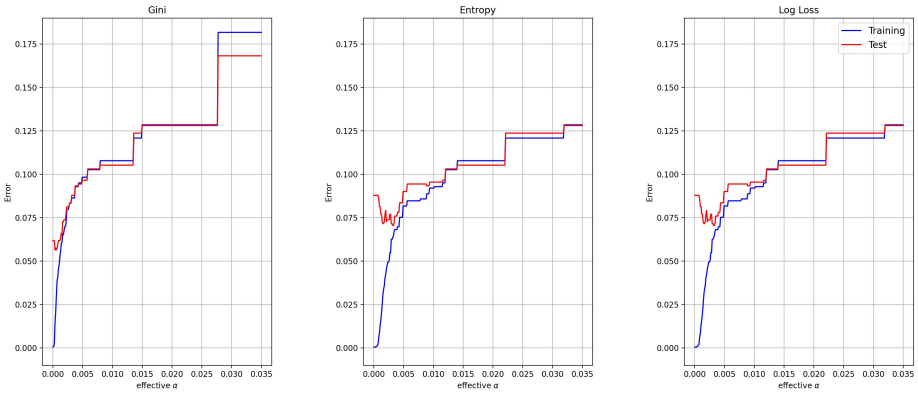

Figure 4: Decision Tree error with cost-complexity pruning

The optimal hyperparameters obtained were 0.0003512 for the Gini impurity and 0.0031605 for the other split criteria. The overall results here appear comparable to the pre-pruning error (Figure 1), as we can see the effect of overfitting for α_{eff} very close to zero. We can also see that very large values of α_{eff} leads to underfitting: the test and training error become larger and approximately equal.

1.2 Random Forests

Random Forests makes classifications based on the average of an ensemble of decision trees trained on a bootstrapped sample of the training set, and a random subset of the features.

1.2.1 Ensemble Size

Figure 6 shows the effect of the ensemble size on the training and testing error. We use the Gini impurity as split criterion for the decision trees as they appeared to give the overall lowest testing error in the previous analyses.

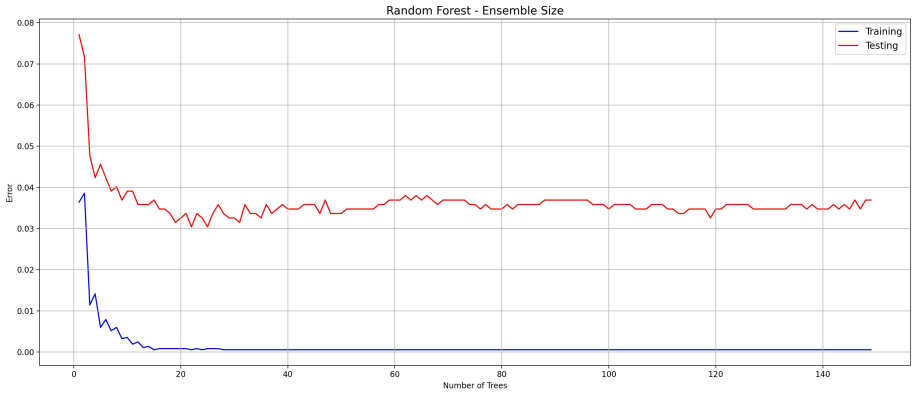

Figure 5: Random Forest error with ensemble size

We can see that both the testing and training error are greatly reduced when using an ensemble of decision trees. The smallest optimal ensemble size here was found to be 22. The test error does not significantly increase or decrease beyond this, suggesting that the ensemble structure of the random forest may make this classifier resistant to overfitting.

1.2.2 Number of Features

A 'rule of thumb' selection for the number of features per tree is generally $\sqrt{d}$ for a dataset of d features. We test this assumption in Figure 6, using the default ensemble size of 50.

The minimal test error is found when 2 features are selected for each decision tree in the ensemble, with marginally better success than $\sqrt{d} \approx 8$ features. Further experimentation would be required to confirm the significance of this.

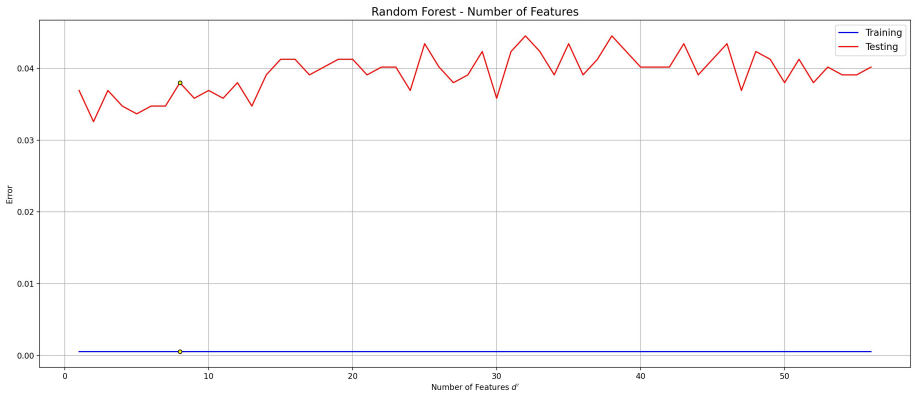

Figure 6: Random Forest error with number of features per tree. The yellow marker indicates the error at $\sqrt{d}$ features.

1.2.3 Sample Size

Finally, we test the error as a function of the training sample size, as shown in Figure 7

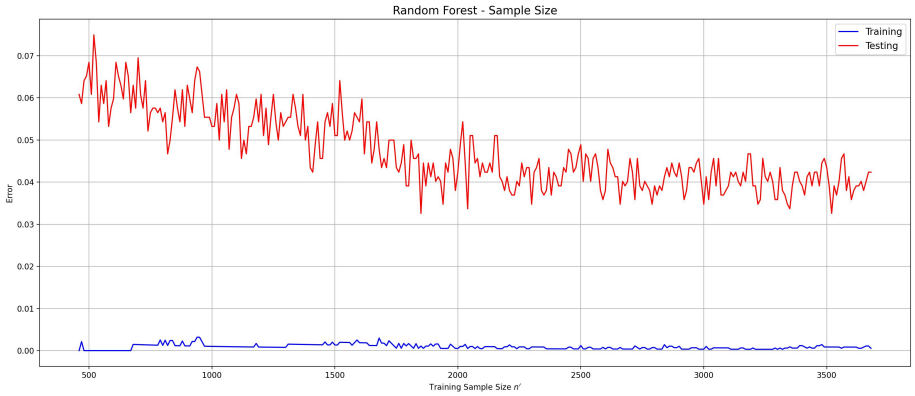

Figure 7: Random Forest error with increasing training sample size.

Similar to the previous analyses for the decision trees, we find the testing error becomes arbitrarily smaller as the training sample size increases. In all cases, however, the test error of the Random Forest model is lower than that of the individual Decision Tree, providing strong evidence towards the efficacy of ensemble models such as this one.

1.3 AdaBoost

The final model under individual investigation is AdaBoost. Similar to Random Forests, AdaBoost is an ensemble model, and produces classifications based on a weighted sum of its individual weak learners. The weak learners in the ensemble are iteratively trained on a distribution of the training set weighted so as to emphasize the mistakes made by the previous learner.

1.3.1 Ensemble Size

As was done with the Random Forest model, we first test the error against the ensemble size. The ensemble consists of decision trees of max depth 1. Figure 9 shows the results:

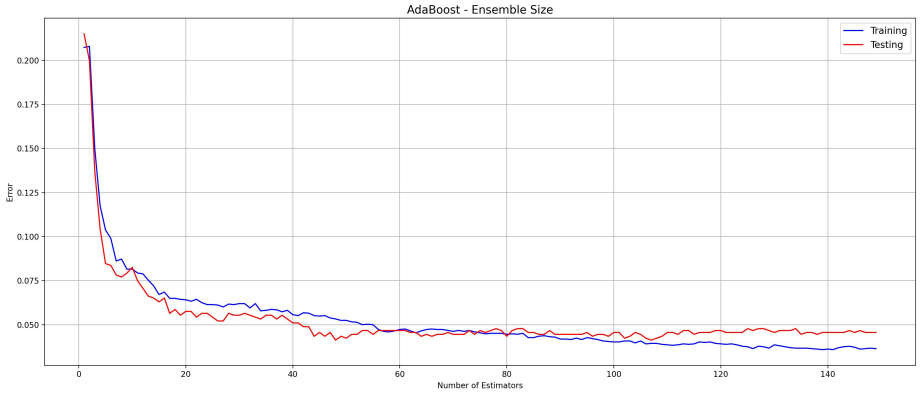

Figure 8: AdaBoost classification error with increasing ensemble size.

The decline of the error is similar to that of the Random Forest model as ensemble size is increased, however the difference between the training and test error is even smaller. This could imply that the generality of the AdaBoost model is more robust than the former, even if the error is not significantly lower. The optimal ensemble size based on these results appears to be 48.

1.3.2 Maximum Depth

Typically AdaBoost weak learners are trained on decision stumps (trees of max depth 1), however we can evaluate the performance using a greater depth. Figure 9 shows the error associated to increasing the depth of the weak learners:

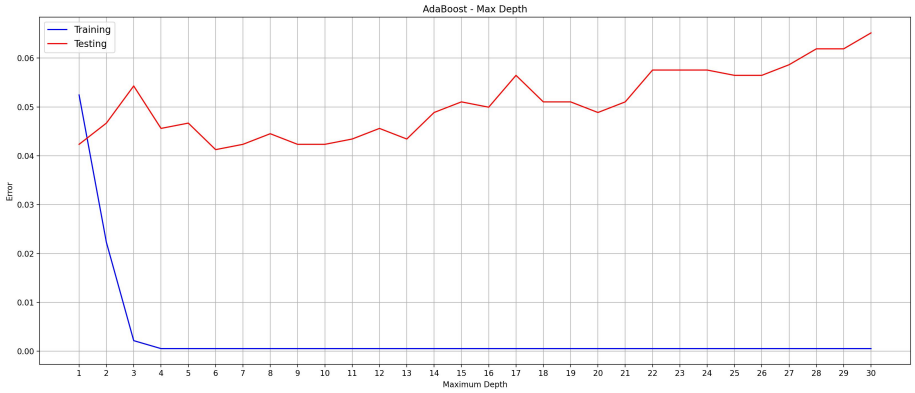

Figure 9: AdaBoost classification error with increasing maximum depth of weak learners.

It is quite apparent from the trends in the error that using learners with very high depths quickly leads to overfitting. The optimal depth was 6 in this analysis, however a max depth of 1 performs nearly as well. Therefore, there is very little evidence to suggest that using decision stumps is worse than using trees of greater depth.

1.3.3 Sample Size

As was done for the previous two models, we end with an analysis of the effect of traning sample size on the error of the AdaBoost model. Figure 10 displays the results:

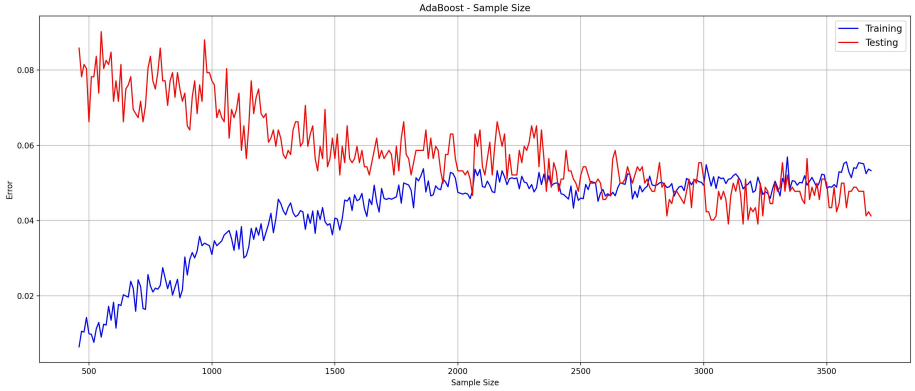

Figure 10: AdaBoost classification error with increasing training sample size.

An interesting result is the overlap of the training and testing error as the training sample approaches the entire dataset. This is consistent with the observations made from Figure 9 during the analysis of the ensemble size; it seems that the AdaBoost model has strong generalizability and is robust to overfitting.

2 Comparative Analysis

2.1 Ensemble Size

The previous analysis suggested that an ensemble size of 22 was best for reducing the test error of the Random Forest, and 48 for AdaBoost. Another stronger determination of the optimal ensemble size is to use cross-validation. Setting the number of features per decision tree to be 2 (determined to be the optimal selection previously) for the Random Forest, and the max depth for the AdaBoost weak learners to 1, We perform cross validation on both models, setting parameter $k = 5$ (five partitions of the training set). Figure 11 shows the validation error for each model:

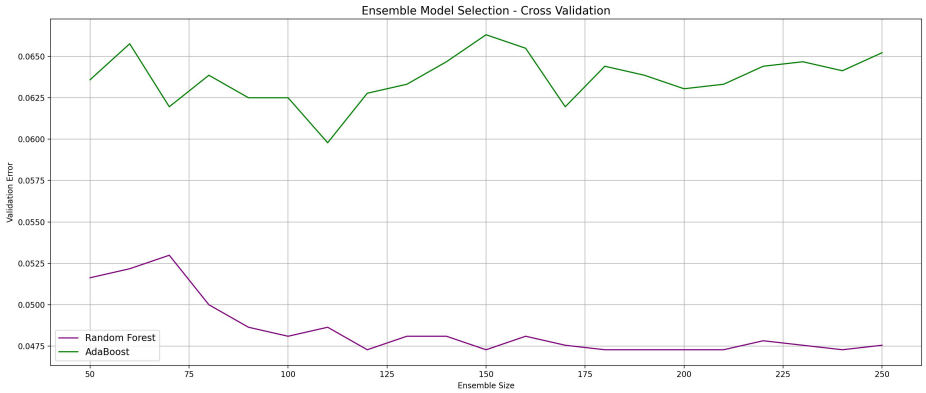

Figure 11: Validation error for AdaBoost and Random Forest models with increasing ensemble size.

Based on these results, we select an ensemble size of **120** for the Random Forest and **110** for AdaBoost.

2.2 Final Results

Using these two optimized models, we determine the final test error of each model. The Random Forest produced a classification error of **0.0036**, and AdaBoost performed with a classification error of **0.0046**. Therefore, we conclude the Random Forest is a superior classifier of the given data. It should be noted that the difference in error between the two models is quite small, and further analyses of other hyperparameters could be performed in future studies to confirm this.

References

[1]F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: machine learning in Python", Journal of Machine Learning Research **12**, 2825–2830 (2011).